

Una guida all'utilizzo di
LogicGateSim.exe
Logic gate sand box

Redatta da Gilles Vismara

Cos'è?

Logic Gate Sim è un programma pensato per poter progettare e testare veloce mente circuiti con le porte logiche. È un programma open source scritto in C#, il download è il codice sorgente si possono trovare a gvs.ovh/cs/lgs.

Istruzioni

Fase di progettazione:

Una volta scaricato ed aperto il file LogicGateSim.exe si presenterà una finestra di dialogo come questa:

Gate:

A questo punto si può iniziare a progettare il proprio circuito, scrivendo con che gate iniziare. I Logic Gates disponibili sono:

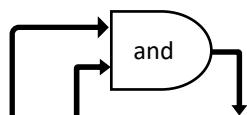
- and: Da output vero solo se entrambi gli input sono veri.
- or: Da output vero se almeno uno degli input è vero.
- xor: Da output vero solo se uno dei due input è vero.
- not: Da output vero solo se l'input è falso.

Come esempio utilizzeremo l'and gate, ma voi potete scegliere quello che più vi aggrada. Una volta qui, bisogna premere il tasto 'invio' e dovremmo trovarci in una situazione di questo tipo:

Gate: and

Input 1:

Prima di passare al prossimo passaggio, dobbiamo comprendere come il programma gestisce gli input e gli output. Immaginiamo una gigantesca griglia formati a da 500 valori. Le porte logiche eseguono le loro operazioni su ognuno di questi valori, quindi essi vanno indicati. Per esempio:



Valori	0	0	0	0	0	0	0	0	0	0	0	0	0
Indici	1	2	3	4	5	6	7	8	9	10	11	12	13

Nell'immagine qui sopra possiamo osservare un and gate che prende come input i valori 0 e 1 dalla griglia, compie l'operazione and e inserisce il risultato nella posizione 2 della griglia. Quindi ora inseriremo i seguenti valori nel programma:

Gate: and

Input 1: 0

Input 2: 1

Output: 2

Gate:

Da questo punto possiamo aggiungere quanti altri logic gates vogliamo, ma per questo primo esempio uno basta. Per terminare dobbiamo dire al programma quali valori mostrare all'utente alla fine della simulazione. Nel nostro caso vogliamo che il programma mostri il valore 2 della griglia:

```
Gate: and
Input 1: 0
Input 2: 1
Output: 2
Gate: out
Input 1: 2
Input 2: -
Output: -
Gate: end
```

Per terminare la fase di progettazione bisogna scrivere in *Gate* 'end'.

Fase di simulazione:

In Questa fase possiamo modificare gli input del nostro circuito. Se tutto è andato correttamente fino ad ora il programma dovrebbe aver già simulato una volta il circuito, quindi ci troviamo in questa posizione:

[...]

```
Gate: end
Pin 2: false
Input:
```

Ora dobbiamo dire al programma quale valore della griglia vogliamo modificare, e che valore vogliamo fargli assumere (*true* o *false*):

[...]

```
Gate: end
Input: 0
Value: true
Input:
```

Possiamo modificare anche altri valori, e quando siamo pronti per testare il circuito di nuovo, possiamo scrivere 'run':

[...]

```
Gate: end
Input: 0
Value: true
Input: run
Pin 2: false
Input:
```

Se si vuole si possono modificare altri valori e continuare a simulare il circuito per quante volte si vuole. Una volta terminato di usare il programma, si può scrivere 'stop' in *input* per chiudere il tutto.

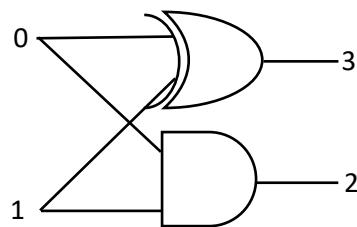
Esempi

Ecco ora alcuni esempi di circuiti con i logic gates da mettere nel programma per prendere confidenza:

Half-adder:

```
Gate: xor
Input 1: 0
Input 2: 1
Output: 2
Gate: and
Input 1: 0
Input 2: 1
Output: 3
Gate: out
Input 1: 2
Input 2: -
Output: -
Gate: out
Input 1: 3
Input 2: -
Output: -
Gate: end
```

Tabella di verità			
Input 0	Input 1	Output 2	Output 3
flase	flase	flase	flase
true	flase	true	flase
flase	true	true	flase
true	true	flase	true



[altri esempi in progettazione]